模块度增量与局部模块度引导下的社区发现算法

刘明阳，张曦煌

(江南大学 物联网工程学院，江苏 无锡 214122)

摘　要：社区结构是复杂网络的重要特性之一，基于层次聚类的社区发现算法很好地利用了模块度来挖掘网络中的社区结构，但其局限性也导致算法对社区结构复杂的网络划分不够准确、无法发现小于一定规模的社区。在层次聚类的基础上，提出引入局部模块度来弥补模块度在划分社区时的不足，避免可能出现的划分不合理情况。通过真实数据集和人工网络进行了验证，实验结果证明，该算法具有可行性与有效性。

关键词：复杂网络；社区发现；层次聚类；模块度增量；局部模块度

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.12.0750

Communities detection during increment of modularity and local modularity

Liu Mingyang, Zhang Xihuang

(*School of Internet of Things Engineering*, *Jiangnan University*, *Wuxi Jiangsu* 214122, *China*)

Abstract: Community structure is one of the important characteristics of Complex Network, the community detection algorithms with hierarchical clustering makes good use of modularity to achieve community detection, but its limitations lead to poor detection quality of networks which have complex structure of community and it may fail to identify communities smaller than a scale. This paper used local modularity to make up the lack of modularity on the basis of hierarchical clustering, it can avoid unreasonable result. The algorithm is tested on real-world data sets and artificial network, experimental results confirm that it has feasibility and effectiveness.

Key words: complex network; community detection; hierarchical clustering; the increment of modularity; local modularity

0　引言

自然界和人类社会中大量的复杂系统都可以被描述成由许多节点和一些连接节点间的边构成的复杂网络，其中节点代表真实系统中不同的个体，而边表示个体之间的关系。例如，社交网络中每个节点代表一个用户，每条边表示两个用户之间存在某种关系。

基于复杂网络的小世界特性和无标度特性，网络中少数节点拥有大量的连边，而大部分节点拥有少量连边，使得网络中的节点呈现出集群特性[1]，而社区发现就是用来发掘网络聚集行为的一种技术。近年来社区发现的研究已经发展为复杂网络研究领域中的重要部分之一，具有广阔的应用前景。例如，在社交网络中，用户与用户之间建立好友关系形成了不同的群体，通过对这些群体中用户之间存在的共性进行分析，进而为信息传播、精准营销、兴趣点推荐等提供宝贵的情报支持。因此，分析复杂网络中承载的社区结构对发现个体的行为规律和实际网络的设计与建设具有重要的指导意义。

1　相关工作

从 2002 年 Newman 和 Girvan 提出网络中存在社区结构开始[1]，发现及分析复杂网络中的社区结构得到了广泛的关注和研究，同时也诞生了许多社区发现算法。目前，主流的社区发现算法类型有谱平分法、标签传播方法、矩阵分块方法以及层次聚类方法等。

社区发现的谱平分法主要包括基于 Laplace 矩阵[2]和基于 Normal 矩阵[3]的方法。基于 Laplace 矩阵的谱平分法把网络的 Laplace 矩阵作为研究对象，Laplace 矩阵的特征值均为实数且总有一个特征值是 0，所以第二小特征值所对应的特征向量为矩阵的第一个非平凡特征向量，该特征向量的元素即为社区划分的依据[4]。基于 Normal 矩阵的谱平分法也是根据特征向量中元素呈现的特征结构来划分社区的。标签传播方法通过模拟信息在节点之间的传播过程生成社区，典型的标签传播算法有 LPA 算法[5]和 SLPA 算法[6]。LPA 算法为所有节点赋予一个唯一的标签，然后统计某个节点所有邻居节点的标签，将出现次数最多的那个标签赋予当前节点，最后将所有拥有相同标签的

收稿日期：2017-12-01；**修回日期：**2018-01-22

作者简介：刘明阳（1994-），男，硕士研究生，主要研究方向为复杂网络（lmy2335@qq.com）；张曦煌（1962-），男，教授，博士，主要研究方向为分布式系统与应用．

节点归并到同一个社区。而 SLPA 算法在 LPA 的基础上引入了 Speaker 和 Listener 两个概念，是对 LPA 算法的一种改进。矩阵分块方法将具有紧密连接关系的节点重新排列，并从中识别出比较稠密的矩阵块[7]，典型的矩阵分块算法有 MB-DSGE 算法。MB-DSGE 算法[8]通过构建一棵完整的层次树，根据密度阈值找到社区结构。层次聚类方法根据网络的层次化特性发现社区，将网络中的某一节点作为初始节点，然后从候选节点的集合中筛选出最合适的节点与初始节点合并，通过这样的方式不断扩展社区规模。Newman 等人引入了模块度(modularity)[9]的概念来评价社区结构划分的质量，自此很多研究者提出了基于模块度的社区发现算法，例如 Newman 提出了基于层次聚类的 Fast-Newman 算法[10]，该算法基于最大化模块度的贪婪思想，合并能够产生最大模块度的两个社区，直到模块度达到最大值为止。Clauset 等人提出了 CNM 算法[11]，优化了节点归并操作的同时采用堆的数据结构，使得算法的复杂度进一步降低。为了有效识别大规模复杂网络的社区结构，研究人员相继开展了关于大规模复杂网络社区发现算法的研究，例如 Clauset[12]提出了基于 CNM 算法的社区并行识别方法，采用并行计算的方式提高了社区合并操作的效率。Riedy 等人[13]提出利用具有多核处理器的服务器并行计算最大模块度的值来进行社区发现等。

自模块度提出以来，将模块度作为划分社区的依据得到了广泛的认可。然而，由于定义本身存在的局限性，使得模块度存在分辨率限制问题。本文根据模块度的优缺点，提出将模块度增量与局部模块度(local modularity)相结合，解决基于模块度的社区发现算法在划分社区时的局限性问题，保证社区发现的准确性与稳定性。

2　模块度增量与局部模块度引导下的社区发现算法

2.1　基本概念

定义 1　社区。在图 $G=(V,E)$ 所定义的网络中，V 表示网络中节点的集合，E 表示网络中边的集合。社区是网络 G 的一个子图，同一社区内节点之间的连接紧密，而不同社区之间节点连接相对稀疏。

定义 2　模块度。为了判断网络的社区结构划分是否具有合理性与有效性，Newman 等人提出了模块度的概念，并将其作为社区划分效果的评价指标。模块度 Q 定义为[9]

$$Q=\sum_{i\in C}\left(\frac{l_i}{m}-\left(\frac{D_i}{2m}\right)^2\right) \tag{1}$$

其中:m 是网络的总边数，l_i 是社区 i 内部总边数，D_i 是社区 i 内部所有节点的节点度之和，节点度即与社区中某一节点相连的边的总数。

模块度的实际物理含义是指网络中社区内部边所占的比例与在同样的社区结构下随机连接社区的边所占的比例的期望值相减得到的差值。Q 的最大值介于 0 到 1 之间，Q 值越大，网络的社区结构越明显，划分效果越好，所以当 Q 达到最大时，说明社区划分达到了最佳效果[9]。然而，判断模块度 Q 是否达到最大值是一个 NP 难题，因此，Newman 提出了模块度增量的概念，其定义为[10]

$$\Delta Q=2(e_{ij}-a_i\times a_j) \tag{2}$$

$$a_x=\frac{\sum_{n\in x}D_n}{2m} \tag{3}$$

其含义是合并社区 i 和社区 j 后产生的模块度变化量 ΔQ，e_{ij} 表示连接 i、j 两个社区的边数所占网络中总边数 m 比例的一半，其中 $i\neq j$，当 i、j 两个社区之间不存在连边时，$e_{ij}=0$。D_n 是社区 x 内节点 n 的节点度，m 为社区 x 的总边数。当 $\Delta Q>0$ 时，说明社区 i 和 j 合并后模块度 Q 增大，反之则减小。

模块度的可靠性得到了广泛的认同，然而，Fortunato 等人[14]提出了模块度在定义上的局限性。他们指出，根据模块度的定义，若包含 l_i 条内部边且内部节点的总节点数是 D_i 的子图满足 $\frac{l_i}{m}-(\frac{D_i}{2m})^2>0$，则这个子图即可视为一个社区。但是在模块度 Q 增大的过程中，若满足此不等式的社区的内部边数 l_i 小于 $\sqrt{2m}$，即使社区间的连接十分稀疏，它们也会被合并成一个大社区[15]。一旦网络中存在许多像这样小于一定规模的社区，那么模块度 Q 的最大值对应的社区划分可能是许多满足模块度定义的小社区的合并，出现远大于其他社区规模的超大社区(Monster)，也就是说，模块度并没有发现并分辨网络中存在的小于一定规模的小社区，有可能出现社区划分结果不合理的情况。

定义 3　局部模块度。在模块度的基础上，Clauset 提出了局部模块度的概念。局部模块度的定义为[12]

$$R=\frac{\sum_{ij}B_{ij}\,\delta(i,j)}{\sum_{ij}B_{ij}}=\frac{I}{T} \tag{4}$$

在图 $G=(V,E)$ 所定义的网络中，社区 C 是网络的一个子图，D 是网络中除社区 C 以外的网络结构，i 和 j 是网络中两个不同的节点，表示节点 i 与 j 之间的连边，则边界邻接矩阵 B_{ij} 定义为

$$B_{ij}=\begin{cases}1; & i<j, i\in C, j\in V, <i,j>\in E\\ 0; & \text{otherwise}\end{cases} \tag{5}$$

当 $i<j$，并且 i 是社区 C 中的节点，节点 j 属于社区 C 或除社区 C 以外的网络结构 D，若节点 i 与 j 之间存在连边，则 $B_{ij}=1$；否则 $B_{ij}=0$。当节点 i 和 j 均属于社区 C 且 i 与 j 之间有连边时，$\delta(i,j)=1$；否则 $\delta(i,j)=0$。所以，局部模块度 R 即为社区内部边的边数 I 与连接社区内节点所有边的边数 T 的比值。一个社区的内部边越多，外部边越少，则该社区的局部模块度 R 值越大，说明社区内部连接越紧密，社区结构越清晰。本文选择以局部模块度为指标抑制大社区进一步扩展，以保证小社区得到更加充分的发展，使得社区发现质量得到一定的提高。

2.2　算法描述

本节对算法的步骤进行详细描述，并提出算法的伪代码。

算法的具体步骤为：

a)参数初始化。在初始化阶段计算出所有算法所需的参数，主要包括网络的节点数n与边数m，以及各个节点的节点度k_x，将每一个节点作为一个社区，根据公式(2)和(3)计算出每个节点对之间的模块度增量ΔQ，并将ΔQ的值存储在一个$n\times n$的数据框(DataFrame)Q_matrix中，构建模块度增量矩阵。例如，节点x与节点y之间的模块度增量的值为ΔQ_{xy}，则将ΔQ_{xy}放入数据框Q_matrix的第x行第y列与第y行第x列中；当$x=y$时，$\Delta Q_{xy}=0$，所以Q_matrix是一个主对角线上的元素等于0的对称矩阵。

b)在所有节点对中查找最大模块度增量和对应的节点组合。在构建模块度增量矩阵的同时，寻找具有最大模块度增量的节点对$<x,y>$。如果发现节点对$<x,y>$中的某个节点与其他多个节点之间具有同样的最大模块度增量，则下一步将这些节点合并为同一个社区，并赋予新社区新的编号。例如，节点对$<i,j>$和节点对$<i,k>$具有相同的最大模块度增量ΔQ_{max}，那么可以先将节点i、j合并为一个新社区l，然后再将节点k与社区l合并。

c)合并社区并更新模块度增量矩阵。文献[11]提出了将两个不同的社区合并后新社区与其他社区之间模块度增量的计算推论：已知合并的两个社区为i和j，它们合并后的新社区编号为$k(k>n)$，新社区k与其他社区h的模块度增量为ΔQ_{kh}，有以下三种情况：

(a)当社区h与社区i、j均有连接或均无连接时，新社区k与社区h的模块度增量为

$$\Delta Q_{kh}=\Delta Q_{ih}+\Delta Q_{jh} \tag{6}$$

(b)当社区h与社区i有连接但与社区j无连接时，代入式(3)，新社区k与社区h的模块度增量为

$$\Delta Q_{kh}=\Delta Q_{ih}-2\times a_j\times a_h \tag{7}$$

(c)当社区h与社区j有连接但与社区i无连接时，代入式(3)，新社区k与社区h的模块度增量为

$$\Delta Q_{kh}=\Delta Q_{jh}-2\times a_i\times a_h \tag{8}$$

可以证明，在第二种情况下，式(7)与式(6)是等价的，证明过程如下：

$$\Delta Q_{kh}=\Delta Q_{ih}-2\times a_j\times a_h=\Delta Q_{ih}+(0-2\times a_j\times a_h)$$

因为社区j和社区h不相连，所以$e_{jh}=0$

$$0-2\times a_j\times a_h=2\times(0-a_j\times a_h)=2\times(e_{jh}-a_j\times a_h)=\Delta Q_{jh}$$

$$\Delta Q_{kh}=\Delta Q_{ih}+\Delta Q_{jh}$$

同理地，在第三种情况下，式(8)和(6)也是等价的。所以，合并的两个社区为i和j，新社区的编号为k，新社区k与其他社区的模块度增量为社区i和社区j与对应社区模块度增量的和。如表1所示，社区组合<2,4>具有最大的模块度增量0.043，则合并社区2和社区4并更新ΔQ矩阵。根据上述证明得到的结论，合并后的结果如表2所示。

如表1、2所示，社区2和社区4合并后产生了新社区5，社区5与社区1、3之间的模块度增量为社区2和社区4与对应社区模块度增量之和，以表2中<5,1>(社区5与社区1)的模块度增量为0.020为例，其为表1中<2,1>的模块度增量0.036和<4,1>的模块度增量-0.016之和。与上述推论相比，在第二和第三中情况下无须计算社区所对应的a_x的值，更新时只需将对应的模块度增量相加，简化了运算过程。

表1　社区合并前的ΔQ矩阵

社区编号	1	2	3	4
1	0	0.036	0.023	-0.016
2	**0.036**	**0**	**-0.015**	**0.043**
3	0.023	-0.015	0	0.030
4	**-0.016**	**0.043**	**0.030**	**0**

表2　合并社区2和4后的ΔQ矩阵

社区编号	1	3	5
1	0	0.023	0.020
3	0.023	0	0.015
5	**0.020**	**0.015**	**0**

d)计算社区的局部模块度。用式(4)(5)计算合并后新社区的局部模块度，在合并的过程中，可能有小部分的节点加入使得局部模块度R下降，但R的整体趋势是逐渐上升的。当社区外的节点x加入时，令原来社区内部边的边数为L_{in}，连接社区内部节点与社区外部节点的边的边数为L_{out}，社区的局部模块度$R=\frac{L_{in}}{L_{in}+L_{out}}$，节点$x$与社区内节点连接的边数为$l_{in}^x$，与社区外节点连接的边数为$l_{out}^x$，则节点$x$加入后，$R$的值为[16]

$$R=\frac{L_{in}+l_{in}^x}{L_{in}+l_{in}^x+L_{out}-l_{in}^x+l_{out}^x}=\frac{L_{in}+l_{in}^x}{L_{in}+L_{out}+l_{out}^x} \tag{9}$$

e)查找与新社区具有最大模块度增量的节点。在矩阵Q新社区所在的行中查找模块度增量的最大值和对应的社区。以表2为例，社区2，4合并后的新社区5与社区1具有最大的模块度增量，则在合并前需要判断社区1是否可以合并到社区5中。

f)判断社区是否可以合并。在合并前，将与新社区具有最大模块度增量的节点$c1$作为社区外的节点通过步骤④计算合并后的局部模块度，并与前一次的值进行比较，如果局部模块度增大，则将节点$c1$合并到社区中，执行步骤c)并返回步骤e)；如果局部模块度减小，则执行步骤c)e)后，令产生最大模块度增量的节点为$c2$，再次通过步骤d)计算社区的局部模块度，如果局部模块度增大，则将$c1$和$c2$合并到社区中，执行步骤c)并返回步骤e)；如果局部模块度连续减小，则说明合并操作已达到社区层次的边界，不能将$c1$和$c2$合并到社区中，社区已经完成合并，执行下一步骤。如果步骤⑤中的最大模块度增量的值小于0，同样将相应节点作为社区外的节点通过步骤d)计算社区的局部模块度，如果局部模块度增大，则将相应节点合并到社区中，执行步骤c)并返回步骤e)；如果局部模块度减小，则不能合并相应节点，社区已完成合并，执行下一步骤。

g)社区完成合并后，将已合并的节点和与其相应的边从网络中移除，余下部分组成新网络，返回步骤 a)，开始发现下一个社区，直到网络中的所有节点都已完成合并，算法结束。

算法的伪代码如下：

```
输入：图 G = (V,E)
输出：社区集合 C
1. n←number_of_nodes(G);//获取网络的节点数
2. length←0;//已合并的节点数
3. while (length < n)
4.  if 未初始化 then
5.     Q _matrix= initialization();//参数初始化，构建模块度//增量矩阵
6.    MC←searchMaxQIndexAndColumn(Q_matrix);//寻找矩//阵中所有拥有最大模块度增量的节点对并存放于队列 MC
7.     c←mergeTwoNode(MC);//将 MC 中的前 2 个节点合并//成新社区 c
8.    length←length + 2;
9.    prior_lm←localModularity(c);//计算新社区 c 的局部模//块度
10.   update(Q_matrix,MC);//更新矩阵和队列 MC
11.  else
12.   MC←searchMaxQIndexInC(Q_matrix);//查找矩阵新社//区所在的行中与新社区有最大模块度增量的节点并加入队//列 MC 中
13.    maxQ←searchMaxQInC(Q_matrix);//查找矩阵新社区//所在的行中最大的模块度增量
14.   now_lm←localModularity(c,x);//计算队列 MC 中的第 //一个节点加入社区时的局部模块度
15.   if now_lm - prior_lm < 0 then
16.     if 局部模块度连续下降 then
17.       c.pop(-1);//删除上一次合并中加入社区 c 的节点
18.       length←length – 1;
19.     end if
20.     if maxq < 0 then
21.       updateGAndC(c);//更新图 G 并将社区 c 加入社区//列表
22.     end if
23.   else
24.     c.insert(x);//将节点 x 并入社区
25.     length←length + 1;
26.     prior_lm = now_lm;
27.     returnToUninitialization();//回到未初始化的状态
28.   end if
29.  end if
30. end while
```

3　实验分析

3.1　数据集描述

为了验证本文提出的社区发现算法具有可行性和有效性，本文采用两种复杂网络数据集进行实验：小规模真实社会网络数据集和使用 LFR benchmar 基准程序[17]生成的人工模拟复杂网络数据集，所有实验均采用 Python 语言实现。

对于小规模真实数据集可以直接将算法的输出结果与真实划分结果对比，但是在很多真实数据集中难以获得真实准确的社区结构信息，因此采用模块度(Q)来评价社区的划分质量；对于较大规模的人工模拟复杂网络数据集主要使用标准互信息(NMI)和综合评价指标(F1Score)来考察算法的划分结果。

3.2　真实社会网络数据集实验

a)Zachary Karate Club[18]是一个著名的小规模真实社会网络数据集，它是基于美国一所大学的空手道俱乐部 34 名成员间的社会关系构建的。网络包含 34 个节点和 78 条边，呈现出分成两个社区的趋势。使用文本的算法对该网络进行划分，实验结果如图 1 所示。

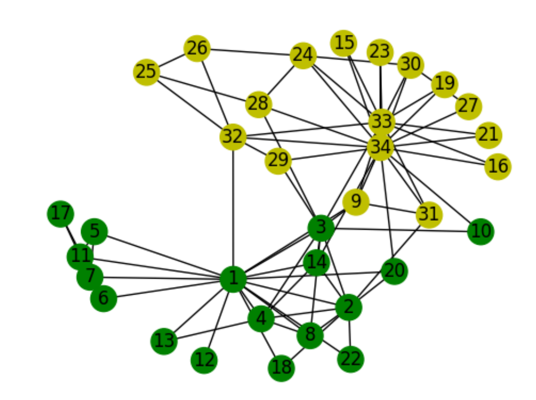


图 1　Zachary Karate Club 的社区发现结果

Karate 网络被划分为两个社区，与真实划分结果相比，所有节点均被准确划分到对应的社区，实验结果良好，证明算法具有初步的可行性与有效性。

b)本文还在其他小规模真实网络数据集上进行了实验，并选用 CNM 算法、SLPA 算法和 Infomap 算法[19]与本文算法(以下简称 Q-lm)进行实验结果的对比。本文实验所用小规模真实网络基本信息如表 3 所示。

表 3　小规模真实网络基本信息

数据集	节点数	边数	备注
dolphins	62	159	新西兰海豚关系网络
lesmis	76	254	小说《悲惨世界》人物关系网络
polbooks	105	441	亚马逊美国政治书籍网络
football	115	616	美国大学生足球联赛网络
jazz	198	2742	欧洲爵士乐音乐家合作网络
neural	291	2148	Elegans 神经网络

四种算法在四种数据集下的实验结果如图 2 所示。

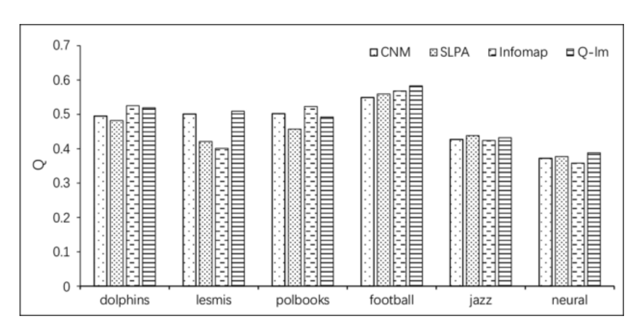


图 2　算法在六种数据集上的模块度 Q 对比

本次实验选取模块度 Q 作为实验结果的评价指标。其中，Infomap 算法是基于信息压缩的随机游走算法，对于所有具有随机性的算法都进行 100 次实验然后将结果取平均值。由图 2 可知，Q-lm 在这六种数据集中都取得了良好的实验效果，并在其中的 football、lesmis 和 neural 网络中得到了最大的模块度 Q。总体而言，Q-lm 算法在部分小规模真实网络下表现出较好的划分效果。

3.3　人工模拟复杂网络数据集实验

LFR benchmar 是由 Lancichinetti 等人提出的一种用于生成人工模拟网络的基准程序，可以根据需求生成相应的人工模拟网络数据集，同时也会生成数据集对应的社区划分结果。本文使用 LFR 基准程序生成的人工网络图基本参数如表 3 所示，其中 N 表示节点数目，k 表示平均节点度，$maxk$ 表示最小节点度，$minc$ 表示规模最小的社区所含的节点数，$maxc$ 表示规模最大的社区所含的节点数，mu(mixing parameter)表示节点与其他社区连接的边数与该节点的度数之间的比值，比值越小，社区之间的界限越清晰，所以 mu 值不宜过大，否则网络的社区结构过于复杂，无法取得良好的划分效果。本次实验的 mu 取值范围是[0.05,0.5]，每次递增 0.05 并生成 10 个其他参数保持一致的网络。

表 4　LFR benchmar 生成的人工网络图基本参数

参数	N	k	maxk	minc	maxc	mu
LFR-10000	10000	20	50	20	100	0.05-0.5

本次实验使用两个评价指标对比分析算法的准确性。第一个指标为标准互信息(NMI)，对于已知划分结果的网络能够很好地评价社区的划分效果，其表达式如下式所示[20]：

$$I(A,B)=\frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}N_{ij}\log\left(\frac{N_{ij}\times N}{N_{i.}\times N_{.j}}\right)}{\sum_{i=1}^{C_A}N_{i.}\log\left(\frac{N_{i.}}{N}\right)+\sum_{j=1}^{C_B}N_{.j}\log\left(\frac{N_{.j}}{N}\right)} \tag{10}$$

其中：C_A 表示社区的标准划分结果，C_B 表示算法的划分结果，i、j 分别为 C_A 和 C_B 中的社区，N 为含混矩阵，矩阵 N 的行对应社区的标准划分结果，矩阵 N 的列对应算法的划分结果，N_{ij} 为 i、j 两个社区中共有的节点数，第 i 行的总和记作 $N_{i.}$，第 j 列的总和记作 $N_{.j}$。NMI 的取值范围在 0 到 1 之间，NMI 的值越大，说明算法的划分结果与标准划分结果越相似。

第二个指标为综合评价指标(F1Score)。F1Score 的定义为准确率(precision)和召回率(recall)的调和平均数，在统计学中用来衡量二分类模型精确度，也可以用来评价社区发现的质量。其定义如下：

$$F1Score=\frac{2\times precision\times recall}{precision\times recall} \tag{11}$$

其中：precision 为分类正确的节点占所属社区中所有节点的比例，recall 为分类正确的节点占真实划分结果的所属社区中所有节点的比例。

本次实验同样选用上述三种算法与 Q-lm 算法进行实验并对比实验结果，由于数据集的节点较多，对于所有具有随机性的算法都进行 10 次实验然后将结果取平均值。四种算法的实验结果如图 3 和 4 所示。

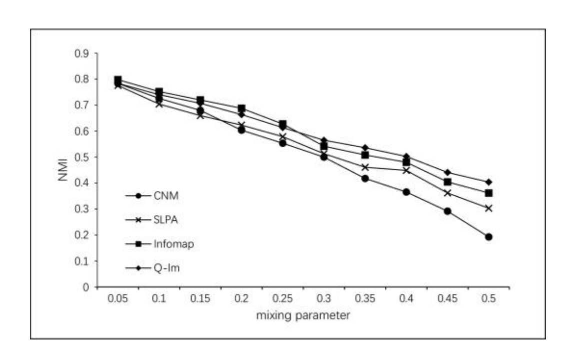


图 3　算法在 LFR-10000 上的 NMI 对比

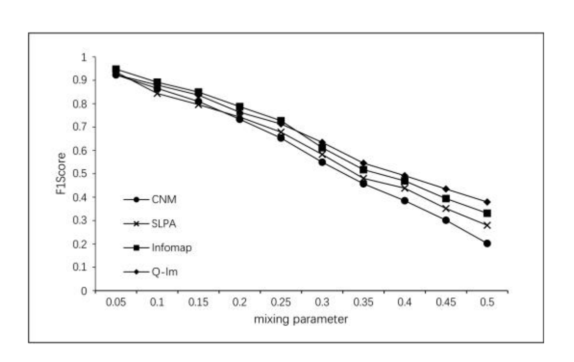


图 4　算法在 LFR-10000 上的 F1Score 对比

由图 3、4 可知，当 mu 值较小时，社区结构清晰，四种算法均能获得良好的划分效果，其中 Infomap 算法的实验效果最好。随着 mu 逐渐增大，社区结构变得模糊，四种算法对应的两种指标都呈现下降趋势，差距也越来越明显。可以看出，这两种指标的变化轨迹大致相似，当 $mu > 0.3$ 时，Q-lm 算法的 NMI 和 F1Score 开始超过 infomap 算法，并在剩余的实验中取得了最好的实验效果，体现了 Q-lm 算法在划分社区边界模糊的网络时具有一定的优势。

通过以上实验，验证了算法引入局部模块度之后在合并节点的过程中能够发现网络中规模较小的社区，有效抑制了社区的过度发展，使得网络中各个社区的规模均处于比较合理的状态，可以得出结论，Q-lm 算法在规模较大、社区结构复杂的网络下表现出了良好的可行性与有效性，相较于其他算法在社区

发现质量方面具有一定的提高。

3.4 算法时间复杂度分析

算法的时间复杂度主要从两个方面考虑：(1)模块度增量矩阵的初始化和更新。(2)社区合并。初始化模块度增量矩阵并寻找最大模块度增量的节点对时，需要遍历一次网络中所有的边，然后对节点对和相应的边进行处理。当识别的社区数目增加、余下的网络规模减小时，初始化矩阵的耗时变得非常小。假设网络中的节点总数为 n，平均每次有 $x(x<n)$个节点需要合并，那么初始化矩阵、寻找节点对并更新矩阵的时间复杂度是 $O(x^2+x)$。生成新社区后，遍历所有与新社区连接的节点寻找节点对并对矩阵进行更新操作，设社区的平均节点数为 m，最坏情况下的时间复杂度是 $O(x^2+x+x\times(m-2)/2)$，综上，忽略掉时间复杂度中较小的部分，算法的总时间复杂度为 $O(nx+n)$。

4 结束语

本文提出的算法基于模块度增量的思想，在社区发现的过程中，结合复杂网络中的相关概念对相应节点和边进行一系列操作，引入局部模块度来控制社区的规模，使得社区划分结果更加合理，并且运行效率良好。通过在真实网络和人工模拟网络下的实验，并与其他算法进行实验结果的对比，证明了算法具有良好的可行性与有效性，尤其在对社区结构复杂的网络进行社区发现时具有比较高的准确度。算法还有很大的提升空间，所以下一阶段的工作是进一步提升算法的性能，提高算法在大规模网络下的运行效率和准确度，使算法在实际网络中具有比较完善的应用价值。

参考文献：

[1] Girvan M, Newman M E J. Community structure in social and biological networks [J]. Proceedings of the National Academy of Sciences of the USA, 2002, 99 (12): 7821–7826.

[2] Newman M E J. Communities, modules and large-scale structure in networks [J]. Nature Physics, 2012, 8 (8): 25-31.

[3] Fortunato S. Community detection in graphs [J]. Physics Reports, 2010, 486 (3–5): 75-174.

[4] Higham D J, Kalnaa G, Milla K. Spectral clustering and its use in bioinformatics [J]. Journal of Computational and Applied Mathematics, 2007, 204 (1): 25-27.

[5] Raghavan U N, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks [J]. Physical Review E, 2007, 76 (3): 036106.

[6] Xie Jierui, Kelley S, Szymanski B K. Overlapping community detection in networks: The state-of-the-art and comparative study [J]. ACM Computing Surveys, 2011, 45 (4): 1-35.

[7] 尚敬文，王朝坤，辛欣，等. 基于深度稀疏自动编码器的社区发现算法 [J]. 软件学报, 2017, 28 (3): 648-662.

[8] Chen J, Saad Y. Dense subgraph extraction with application to community detection [J]. IEEE Trans on Knowledge & Data Engineering, 2012, 24 (7): 1216-1230.

[9] Newman M E J, Girvan M. Finding and evaluating community structure in networks [J]. Physical Review E, 2004, 69 (2): 026113.

[10] Newman M E J. Fast algorithm for detecting community structure in networks [J]. Physical Review E, 2004, 69 (6): 066133.

[11] Clauset A, Newman M E J, Moore C. Finding community structure in very large networks [J]. Physical Review E, 2004, 70 (6): 066111.

[12] Clauset A. Finding local community structure in networks [J]. Physical Review E, 2005, 72 (2): 026132.

[13] Riedy J, Bader D A, Meyerhenke H. Scalable Multi-threaded Community Detection in Social Networks [C]// Proc of IEEE International Parallel and Distributed Processing Symposium Workshops & Phd Forum. Washington DC: IEEE Computer Society, 2012: 1619-1628.

[14] Fortunato S, Barthelemy M. Resolution limit in community detection [J]. Proceedings of the National Academy of Sciences of the USA, 2007, 104 (1): 36–41.

[15] 张聪，沈惠璋. 基于谱方法的复杂网络中社团结构的模块度 [J]. 系统工程理论与实践, 2013, 33 (5): 1231-1239.

[16] 牛冬冬，陈鸿昶，于洪涛，等. 基于局部模块度的社区层次结构发现方法 [J]. 信息工程大学学报, 2013, 14 (3): 364-370.

[17] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms. [J]. Physical Review E, 2008, 78 (4): 046110.

[18] W W Zachary. An information flow model for conflict and fission in small groups [J]. Journal of Anthropological Research 33, 1977, 452-473.

[19] Rosvall M, Bergstrom C T. Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated Systems [J]. PLOS One, 2011, 6 (4): e18209.

[20] Danon L, Diaz-Guilera A, Duch J, *et al.* Comparing community structure identification [J]. Journal of Statistical Mechnics Theroy and Experiment, 2005, 2005 (9): P09008.